````
---
title: "Assignment5"
author: "Ram"
date: "11/8/2021"
output: word_document
---
````

Load libraries required

````
```{r}

library(Benchmarking)
library(lpSolveAPI)

```
````

````
```{r}

DMU1<- read.lp("C:/Users/ramne/Desktop/QMM
Assignment/Assignment5/DMU1.lp")
DMU1
solve(DMU1)
get.objective(DMU1)
get.variables(DMU1)

```
````

The lp acheives maximum efficiency 1 for DMU1.

Given inputs and outputs when we use the weights 5.17 and
1.12 for the outputs, 7.14 and 0.00 for the input for maximum efficiency.

````
```{r}

DMU2<- read.lp("C:/Users/ramne/Desktop/QMM
Assignment/Assignment5/DMU2.lp")
DMU2
solve(DMU2)
get.objective(DMU2)
get.variables(DMU2)

```
````

The lp acheives maximum efficiency 1 for DMU2.

Given inputs and outputs when we use the weights 1.29 and
6.8 for the outputs, 0.00 and 4.7 for the input for maximum efficiency.

````
```{r}

DMU3<- read.lp("C:/Users/ramne/Desktop/QMM
Assignment/Assignment5/DMU3.lp")
DMU3
solve(DMU3)
get.objective(DMU3)
get.variables(DMU3)
````
````

```
The lp acheives maximum efficiency 1 for DMU3.

Given inputs and outputs when we use the weights 1.7 and
3.7 for the outputs, 2.3 and 0.00 for the input for maximum efficiency.

```{r}

DMU4<- read.lp("C:/Users/ramne/Desktop/QMM
Assignment/Assignment5/DMU4.lp")
DMU4
solve(DMU4)
get.objective(DMU4)
get.variables(DMU4)

```
The lp acheives efficiency 0.98 with DMU4.

Given inputs and outputs when we use the weights 1.9 and
0.0 for the outputs, 1.05 and 1.63 for the input for maximum efficiency.
Even though we provide the greatest weight to deposits, DMU4 is not
efficient.

```{r}

DMU5<- read.lp("C:/Users/ramne/Desktop/QMM
Assignment/Assignment5/DMU5.lp")
DMU5
solve(DMU5)
get.objective(DMU5)
get.variables(DMU5)

```
The lp acheives efficiency 0.96 for DMU5.

Given inputs and outputs when we use the weights 1.03 and
5.3 for the outputs, 1.11 and 2.99 for the input for maximum efficiency.
Even though we provide the greatest weight to deposits, DMU5 is not
efficient.

```{r}
DMU6<- read.lp("C:/Users/ramne/Desktop/QMM
Assignment/Assignment5/DMU6.lp")
DMU6
solve(DMU6)
get.objective(DMU6)
get.variables(DMU6)

```
The lp acheives efficiency 0.86 for DMU6.

Given inputs and outputs when we use the weights 1.46 and
7.56 for the outputs, 1.59 and 4.26 for the input for maximum efficiency.
```

Even though we provide the greatest weight to deposits, DMU6 is not efficient.

Let's define our inputs and outputs as vectors .There are 2 inputs (Staff hours, Supplies) and 2 outputs(Ã¢â,¬Å“Reimbursed Patient_DaysÃ¢â,¬Â,"Privately Paid Patient_Day)

```{r}

x <- matrix(c(150, 400, 320, 520, 350, 320, 0.2, 0.7, 1.2, 2.0, 1.2, 0.7), ncol = 2)
y <-
matrix(c(14000,14000,42000,28000,19000,14000,3500,21000,10500,42000,25000,15000),ncol = 2)
colnames(x) <- c("Staff_Hours", "Supplies")
colnames(y) <- c("Reimbursed Patient_Days","Privately Paid Patient_Days")
print(x)
print(y)
Matrix<- cbind(x,y)
row.names(Matrix) = c("Faci1", "Faci2", "Faci3", "Faci4", "Faci5", "Faci6")
Matrix

```

1) Formulate and perform DEA analysis under all DEA assumptions of FDH, CRS, VRS,
IRS, DRS, and FRH.

```{r}

#Free disposability hull
FDH <- dea(x,y, RTS = "fdh")
FDH
peers(FDH)
FDH_Weights <- lambda(FDH)

```
The peer for each facility is same as the peer.

```{r}

#Constant returns to scale, convexity and free disposability

CRS <- dea(x,y, RTS = "crs")
CRS
#Identify Peers
peers(CRS)
#Identify lambda
CRS_Weights <- lambda(CRS)

```
The results show DMU 1,2,3,4 are efficient and DMU 5 is 0.9775, DMU 6 0.867
The peer for 5 and 6 are 1,2,3

```{r}
#Variable returns to scale, convexity and free disposability
VRS <- dea(x,y, RTS = "vrs")
VRS
peers(VRS)
VRS_Weights <- lambda(VRS)
```
All facilities are efficient except DMU5 which is 0.8963

```{r}

#Increasing returns to scale, (up-scaling, but not down-scaling), convexity and free disposability
IRS <- dea(x,y, RTS = "irs")
IRS
peers(IRS)
IRS_Weights <- lambda(IRS)

```

Decreasing returns to scale, convexity, down-scaling and free disposability

```{r}


DRS <- dea(x,y, RTS = "drs")
DRS
peers(DRS)
DRS_Weights <- lambda(DRS)

```

```{r}

FRH <- dea(x,y, RTS="add")
FRH
peers(FRH)
FRH_Weights <- lambda(FRH)

```

```{r}

as.data.frame(Matrix)
DataFrame<- data.frame(CRS = c(1.0000, 1.0000, 1.0000, 1.0000, 0.9775, 0.8675), FDH = c(1, 1, 1 ,1 ,1 ,1), VRS =c(1.0000, 1.0000, 1.0000, 1.0000 ,1.0000, 0.8963), IRS = c(1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 0.8963), DRS = c(1.0000, 1.0000, 1.0000 ,1.0000 ,0.9775, 0.8675), FRH = c(1, 1, 1 ,1, 1, 1))
DataFrame

```
So, from the above results,

1.Facilities 1,2,3,4 are fully efficient for all the assumptions and Facilities 5,6 are not efficient.

2.Facility 5 is fully efficient for FDH, VRS, IRS and FRH assumptions.
3.It is observed that 97.7% efficient for CRS and DRS assumptions.
4.Facility 6 is fully efficient for FDH and FRS assumptions.
5.For the Facility 6, CRS and DRS assumptions 86.7% efficient.
6.For the Facility 6, IRS and VRS assumptions 89.6% efficient.

Question 2 : GOAL PRORAMMING

Maximize Z = P - 6C - 3D, where
P = total (discounted) profit over the life of the new products,
C = change (in either direction) in the current level of employment,
D = decrease (if any) in next yearÃ¢â‚¬â„¢s earnings from the current yearÃ¢â‚¬â„¢s level.
Profit P is expressed as:
P = 20x1 +15x2 + 25x3
Employment level is expressed as :
6x1 + 4x2 + 5x3 = 50
Next year Earnings goal is expressed as:
8x1 + 7x2 + 5x3 >=75
1) Model_Formulation:
Let us consider
y1 - Employment Level minus the target and
y2 - Next Year Earnings minus the Target
y1+ - Penalty for employment level goal exceeding 50
y1- - Penalty for employment level goal decreasing below 50
y2+ - Exceed the next year earnings
y2- - Penalty for not reaching the next year earnings
y1 = 6x1 + 4x2 + 5x3 - 50
y2 = 8x1 + 7x2 + 5x3 -75
For Employment level goal
y1 = y1+ - y1- where y1+, y1- >=0
y1+ - y1-= 6x1 + 4x2 + 5x3 - 50
For Next year earnings goal
y2 = y2+ - y2- where y2+, y2- >=0
y2+ - y2- = 8x1 + 7x2 + 5x3 -75
Final Formulation is expressed as
Max P = 20x1 +15x2 + 25x3
6x1 + 4x2 + 5x3 - (y1+ - y1-) = 50
8x1 + 7x2 + 5x3 - (y2+ - y2-) = 75
xj >=0, where j=1,2,3
yi + >=0, where i= 1,2
yi - >=0, where i= 1,2

2)Managements objective function Objective Function

Maximize Z = P - 6C - 3D
Objective function in terms of x1, x2, x3, y1+, y1- , y2+ and y2-
Max Z = 20x1 +15x2 + 25x3 - 6y1+ - 6y1- - 3y2-
S.T.:

6x1 + 4x2 + 5x3 - y1+ + y1- = 50
8x1 + 7x2 + 5x3 - y2+ + y2- = 75
xj >=0 where j=1,2,3
yi + >=0 where i= 1,2
yi - >=0 where i= 1,2

3) Formulate and solve the linear programming model

```{r}

DEA<- read.lp("C:/Users/ramne/Desktop/QMM
Assignment/Assignment5/Emax.lp")
DEA
solve(DEA)
get.objective(DEA)
get.variables(DEA)
get.constraints(DEA)



```

From the above result, penalty for not satisfying the goals on the objective function is 225.
The order shows the order in which the variables were written in the objective function. The
results show that x1 = 0, x2 = 0, x3 = 15, y1+ = 25, y1Ã¢Ë†â€™ = 0, y2+ = 0, y2Ã¢Ë†â€™ = 0, which indicates that the Next years Earnings (y2) expectations are fully satisfied, but the
Employment level goal is exceeded by 25 with the total profit of product 3, there is a
negative result on its profit by 15.